First Hit Fwd Refs

□ Generate Collection

L15: Entry 2 of 19 File: USPT Jan 27, 2004

DOCUMENT-IDENTIFIER: US 6684244 B1
TITLE: Aggregated policy deployment and status propagation in network management systems

Application Filing Date (1):
20000107

Brief Summary Text (7):
Deploying policy involves moving the policy onto a target or policy configuration agent, translating the policy into target-specific configuration, and loading the configuration. If performed one at a time for each target, this process can be complicated and time consuming. In addition, there can be a significant time delay in deploying related policies to one or more targets. Complicating this situation is the fact that deployment of a given policy to its target does not always occur in the order that the policy was moved by the server program and, in fact, implementation on the target may not actually occur due to errors or inconsistencies in the policy. Thus, confusing and conflicting situations can exist when policies are individually deployed. In addition, with multiple targets receiving policies at differing times, it has been impossible to keep track of which targets have actually implemented the policy changes.

Brief Summary Text (12):
As referred to herein, a target is a process or resource that is being managed using policy. The managed item itself may be able to recognize and conform to the policy directly, or may be managed by a proxy which recognizes policy information and converts it to configuration information that the managed entity can recognize and conform to.

Detailed Description Text (7):
FIG. 1 is a drawing showing a policy 120 related to a target 110 as described in various representative embodiments of the present patent document. As referred to herein, the target 110 is a process or resource that is being managed using policy 120. The managed item itself may be able to recognize and conform to the policy 120, or may be managed by a proxy which recognizes policy 120 information and converts it to configuration information that the managed entity can recognize and conform to. In representative embodiments, the present patent document discloses techniques by which multiple policies 120 can be deployed in groups in order to manage separate aspects of specified devices, i.e., targets 110.

First Hit

□ Generate Collection

L15: Entry 1 of 19 File: PGPB Jun 19, 2003

DOCUMENT-IDENTIFIER: US 20030115246 A1
TITLE: POLICY MANAGEMENT FOR HOST NAME MAPPED TO DYNAMICALLY ASSIGNED NETWORK ADDRESS

Application Filing Date:
19990824

Summary of Invention Paragraph:
[0008] As referred to herein, a target is a process or resource that is being managed using a policy or policies. The managed item itself may be able to recognize and conform to the policy, or may be managed by a proxy which recognizes policy information and converts it to configuration information that the managed entity can recognize and conform to.

Detail Description Paragraph:
[0029] As referred to herein, a target is a process or resource that is being managed using a policy or policies. The managed item itself may be able to recognize and conform to the policy, or may be managed by a proxy which recognizes policy information and converts it to configuration information that the managed entity can recognize and conform to.

First Hit Fwd Refs

□ Generate Collection

L15: Entry 3 of 19 File: USPT Aug 26, 2003

DOCUMENT-IDENTIFIER: US 6611863 B1
TITLE: Automatic device assignment through programmable device discovery for policy based network management

Application Filing Date (1):
20000605

Detailed Description Text (9):
Therefore, according to an example embodiment, device proxies 116 can receive a policy from the policy server 112, convert the policy to a device-specific configuration (i.e., a configuration that is native to the device) and then distribute the policy to one or more devices 120 within network 100 using native or device-specific communication protocols. Each proxy may perform these functions for one or more received policies and for one or more groups of policy managed devices.

Detailed Description Text (10):
In the above described Cisco/Intel example, device proxy 116A would receive a first policy from the policy server 112 using a common protocol, such as COPS (or other protocol), convert the first proxy to a first device-specific configuration (e.g., a configuration that is native to these Cisco routers), and then distribute the policy to the Cisco routers (120A-120C) using a device-specific communications protocol, such as SNMP in this example. Likewise, the device proxy 116Z will receive a second policy from the policy server 112 using COPs (or other protocol), convert the second policy to a second device-specific configuration (e.g., a configuration that is native to the Intel NICs) and then send (or distribute) the second policy to the Intel NICs (120D-120E) using a device-specific communications protocol that is specific or native to the Intel NICs (such as Distributed Component Object Model or DCOM in this example).

CLAIMS:

22. The method of claim 21, wherein the proxy policy managing comprising: receiving a policy; converting the policy to a device-specific configuration; and distributing the policy to one or more matching devices using a device-specific protocol.

First Hit Fwd Refs

□ Generate Collection

L15: Entry 4 of 19 File: USPT Jul 1, 2003

DOCUMENT-IDENTIFIER: US 6587876 B1
TITLE: Grouping targets of management policies

Application Filing Date (1):
19990824

Brief Summary Text (11):
As referred to herein, a target is a process or resource that is being managed using a policy or policies. The managed item itself may be able to recognize and conform to the policy, or may be managed by a proxy which recognizes policy information and converts it to configuration information that the managed entity can recognize and conform to.

Detailed Description Text (8):
As referred to herein, a target is a process or resource that is being managed using a policy or policies. The managed item itself may be able to recognize and conform to the policy, or may be managed by a proxy which recognizes policy information and converts it to configuration information that the managed entity can recognize and conform to.

WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Monday, April 26, 2004

Hide?	Set Name	Query	Hit Count
		DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ	
□	L15	l13 NOT l14	19
□	L14	L13 and command	4
□	L13	20000712	23
□	L12	(policy or policies or QoS or quality) near8 (transform or transforming or translate or translating or convert or converting) near8 (configurate or configuration or configurating)	61
□	L11	20000712	3
□	L10	command near8 (represent or representation) near8 (policy or policies)	5
□	L9	(merging or aggregation) near8 command near8 (state adj3 value)	0
□	L8	configuration near8 (policy or policies or QoS) near8 CLI	4
□	L7	(merge or merging or unify or unifying) near8 configuration near8 (policy or policies or QoS)	1
□	L6	L5 and l4	0
□	L5	(policy or policies or QoS) near8 (parse or parsing or transform or transforming or translate or translating) near8 (command or object or basic)	37
□	L4	20000712	166
□	L3	configuration near8 (parse or parsing or transform or transforming or translate or translating) near8 (command or object or basic)	279
□	L2	20000712	2
□	L1	(current or existing) near8 configuration near8 (parse or parsing or transform or transforming or translate or translating) near8 (command or object or basic)	5

END OF SEARCH HISTORY

First Hit Fwd Refs

□ Generate Collection

L15: Entry 6 of 19 File: USPT Oct 19, 1999

DOCUMENT-IDENTIFIER: US 5968176 A
TITLE: Multilayer firewall system

Application Filing Date (1):
19970529

Brief Summary Text (23):
Thus the present invention can be characterized according to one aspect as a system that provides security in a network including nodes. Nodes in a set of the nodes in the network include security functions operating in one or multiple protocol layers, and execute such security functions in response to configuration data having formats adapted for the respective types of nodes. The system includes a topology data store, that stores information about security functions operating in the set of nodes in the network, and about interconnection of nodes in the network. A configuration interface is coupled to the topology data store. The interface includes an input by which to receive security policy statements indicating security policies to be implemented among nodes in the network. A configuration driver is coupled to the network, the configuration interface and the topology data store. The configuration driver includes resources which translate the security policy statements into configuration data in the formats needed for nodes in the network, and which send the configuration data to the nodes using the communication channels available for the respective nodes.

Brief Summary Text (24):
According to various aspects of the invention, the nodes execute multiple protocol layers, including a medium access control MAC layer, and the set of nodes includes nodes that provide MAC layer filtering according to filter parameters. The configuration data includes filter parameters for the MAC layer filtering. In another aspect, the multiple protocol layers include a network layer, such as an Internet Protocol IP layer. The set of nodes according to this aspect includes nodes providing network layer filtering according to filter parameters. The configuration data includes filter parameters for the network layer filtering in such nodes. According to another aspect, the multiple protocol layers include a transport layer function, such as the Transport Control Protocol TCP operating over the Internet Protocol IP. According to this aspect, the configuration driver includes resources that translate security policy statements into configuration data for the transport layer functions, such as filtering, application layer functions such as filitering, and /or for functions in higher layers of the protocol stack. Such higher layer functions include for example authentication protocols, authorization protocols, auditing protocols and other security functions. A variety of devices executing filtering, access control, protected communications and security assist features are distributed in the network infrastructure, and managed in a coordinated fashion according to the present invention.

Brief Summary Text (30):
The present invention can also be characterized more generally as a method for establishing a firewall system in a network. The method includes providing topology data including information about security functions operating in nodes in the network, and about interconnection of nodes in the network. Next, the method

includes providing security policy statements including security policies to be implemented among or between end systems in the secured network, using formats and communication channels matched to the type or types of nodes involved. Next, the method involves translating in response to the topology data, the security policy statements into configuration data for security functions operating in the network. Finally, the method includes establishing the configuration data in the security functions at the active nodes in the network, using formats and communication channels matched to the various type or types of nodes. The multiple layers of protocol at which the security functions operate in one alternative include at least two protocol layers, for example at least two of the data link layer, network layer, transport layer, and applications or equivalents thereof.

Detailed Description Text (5):
The configuration interface front end 31 is coupled with the topology database 30. It includes inputs by which to receive security policy statements, such as by providing a script in a security policy language, which is interpreted by an interpreter 34 to provide security policy statements. The security policy management back end 32 is coupled to the configuration interface front end 31 and to the topology database 30, and includes resources that translate the security policy statements into configuration data for nodes in the network. The security policy management back end 32 provides a configuration driver that establishes the configuration data to the security policy management agents 22-26 at nodes in the network in which the security policy statements are to be enforced.

Detailed Description Text (18):
The security policy management back end uses the information from the security policy front end configuration interface and the topology data base to create, store, update, distribute and enforce the security policy specified by the security policy statements. The back end consists of elements in stand alone management systems, in persistent storage systems and in nodes. Security policy management back end translates the rules specified in the security policy statements in a context of the information in the topology data base and creates node specific security policy configuration data that it distributes to the network nodes it has chosen. The security policy management back end decides how to partition the security policy statements into sets of configuration data enforceable at specific nodes, and transforms the rules of the security policy statements into node specific configuration data enforceable at the chosen nodes.

Detailed Description Text (27):
For each of these active nodes, translate the security policy statement specified in the rule into security policy configuration data that the node can enforce, i.e., rules in its own security policy language.

CLAIMS:

1. A system providing multiple protocol layer security in a network including nodes of a plurality of network device types, with nodes in a set of the nodes in the network including security functions executing in response to configuration data adapted for the corresponding type of node in the network, comprising:

a topology data store, storing information about security functions operating in the set of the nodes in the network, and about interconnection of nodes in the set of the nodes in the network;

a configuration interface, coupled to the topology data store, including an input by which to receive security policy statements indicating security policies to be implemented among nodes in the network; and

a configuration driver, coupled to the network, the configuration interface, and the topology data store, including resources which translate the security policy statements into configuration data for the plurality of types of nodes in the network, and which conveys the configuration data to the nodes, wherein the security functions operating in the plurality of network device types across multiple protocol layers are coordinated by the security policy so that particular device types enforce the part of the security policy pertinent to the associated part of the network.

21. A system providing security in a network including nodes of a plurality of types, nodes in a set of the nodes in the network including security functions executing in response to configuration data adapted for the corresponding type of node, comprising:

a topology data store, storing information about security functions operating in the set of the nodes in the network, and about interconnection of nodes in the set of the nodes,

wherein the topology data store includes data indicating nodes coupled to network links to nodes external to the set of nodes, active nodes in the network capable of enforcing a security policy and passive nodes which are incapable of enforcing, or not trusted to enforce, a security policy; and

wherein the security policy statements indicate security policies for active nodes, passive nodes, and for communications traversing network links to nodes external to the set of the nodes in the network;

a configuration interface, coupled to the topology data store, including an input by which to receive security policy statements indicating security policies to be implemented among nodes in the network; and

a configuration driver, coupled to the network, the configuration interface, and the topology data store, including resources which translate the security policy statements into configuration data for the plurality of types of nodes in the network, and which conveys the configuration data to the nodes.

22. A system providing security in a network including nodes of a plurality of types, nodes in a set of the nodes in the network including security functions executing in response to configuration data adapted for the corresponding type of node, comprising:

a topology data store, storing information about security functions operating in the set of the nodes in the network, and about interconnection of nodes in the set of the nodes, wherein the topology data store includes data indicating active nodes capable for enforcing a security policy and passive nodes which are incapable of enforcing, or not trusted to enforce, a security policy;

a configuration interface, coupled to the topology data store, including an input by which to receive security policy statements indicating security policies to be implemented among nodes in the network; and

a configuration driver, coupled to the network, the configuration interface, and the topology data store, including resources which translate the security policy statements into configuration data for the plurality of types of nodes in the network, and which conveys the configuration data to the nodes.

25. A system providing security in a network including nodes of a plurality of types, nodes in a set of the nodes in the network including security functions executing in response to configuration data adapted for the corresponding type of node, comprising:

a topology data store, storing information about security functions operating in the set of the nodes in the network, and about interconnection of nodes in the set of the nodes;

a configuration interface, coupled to the topology data store, including an input by which to receive security policy statements indicating security policies to be implemented among nodes in the network, wherein the configuration interface includes a script interpreter which interprets a script language to determine the security policy statements, wherein the script language includes a syntax for specifying a security policy statement including a source set identifier, a destination identifier, a communication activity identifier, and a rule for the identified communication activity between the identified source set and the identified destination set; and

a configuration driver, coupled to the network, the configuration interface, and the topology data store, including resources which translate the security policy statements into configuration data for the plurality configuration data to the network, and which conveys the configuration data to the nodes, wherein the configuration driver includes resources to identify security policy statements which cannot be enforced according to the data in the topology data store.

26. A system providing security in a network including nodes of a plurality of types, nodes in a set of the nodes in the network including security functions executing in response to configuration data adapted for the corresponding type of node, comprising:

a topology data store, storing information about security functions operating in the set of the nodes in the network, and about interconnection of nodes in the set of the nodes, wherein the topology data store includes data structures providing information for particular nodes, including network layer addresses, medium access control MAC layer addresses, user identifiers, whether or not the particular node is trusted to enforce security policy, the type of security policy it is able to enforce, and its connections to other nodes;

a configuration interface, coupled to the topology data store including an input by which to receive security policy statements indicating security policies to be implemented among nodes in the network; and

a configuration driver, coupled to the network, the configuration interface, and the topology data store, including resources which translate the security policy statements into configuration data for the plurality of types of nodes in the network, and which conveys the configuration data to the nodes.

27. A system providing security in a network including nodes of a plurality of types, nodes in a set of the nodes in the network including security functions executing in response to configuration data adapted for the corresponding type of node, comprising:

a topology data store, storing information about security functions operating in the set of the nodes in the network, and about interconnection of nodes in the set of the nodes;

a configuration interface, coupled to the topology data store, including an input by which to receive security policy statements indicating security policies to be implemented among nodes in the network, wherein the security policy statements indicate security policies for communication between a source set including one or more end stations in the network, and a destination set including one or more end stations in the network, and wherein the configuration driver includes resources to identify a cut vertex set of nodes capable of enforcing the indicated security policies within the set of nodes in the network, and to establish the configuration data in the nodes in the cut vertex set; and a configuration driver, coupled to the network, the configuration interface, and the topology data store, including resources which translate the security policy statements into configuration data for the plurality of types of nodes in the network, and which conveys the configuration data to the nodes.

29. A system providing security in a network including nodes of a plurality of types, nodes in a set of the nodes in the network including security functions executing in response to configuration data adapted for the corresponding type of node in the network, comprising:

a topology data store, storing information about security functions in the set of the nodes in the network, and about interconnection of nodes in the set of the nodes in the network, the topology data store including data structures providing information for particular nodes, including addresses at one or more protocol layers, whether or not the particular node is trusted to enforce security policy, the type of security policy the particular node is able to enforce, and connections of the particular node to other nodes;

a configuration interface, coupled to the topology data store, including an input by which to receive security policy statements indicating security policies to be implemented between source sets of one or more end stations and destination sets of one or more end stations in the network, including a script interpreter which interprets a script language to determine the security policy statements, and the script language includes a syntax for specifying a security policy statement including a source set identifier, a destination set identifier, a communication activity identifier, and a rule for the identified communication activity between the identified source set and the identified destination set; and

a configuration driver, coupled to the network, the configuration interface, and the topology data store, including resources which translate the security policy statements into configuration data for various types of nodes in the network, and which send the configuration data to the nodes.

43. A method for establishing a firewall system in a network including a set of nodes of a plurality of types, nodes in the set of nodes in the network including security functions executing in response to configuration data adapted for the corresponding node, comprising:

providing topology data including information about security functions operating in nodes in the set, and about interconnection of nodes in the set,

providing security policy statements indicating security policies to be implemented among end systems in the set;

translating, in response to the topology data, the security policy statements into configuration data for security functions operating at nodes in the set; and

establishing the configuration data in the security functions at the nodes in the network;

wherein the topology data includes data structures providing information for particular nodes, including addresses at one or more protocol layers, whether or not the particular node is trusted to enforce security policy, the type of security policy the particular node is able to enforce, and connections of the particular node to other nodes.

52. The method of claim 49, wherein the step of translating includes, to enforce security policies for passive nodes, generating configuration data for active nodes linked to passive nodes.

63. A method for establishing a firewall system in a network including a set of nodes of a plurality of types, nodes in the set of nodes in the network including security functions executing in response to configuration data adapted for the corresponding type of node in the network, comprising:

providing topology data including information about security functions operating in nodes in the set, and about interconnection of nodes in the set;

providing security policy statements indicating security policies to be implemented between a source set of end stations and a destination set of end stations in the set;

identifying, in response to the topology data and the security policy statements, a cut vertex set of nodes consisting of nodes capable of enforcing the security policy statements, and which if removed from the network would isolate the source set from the destination set;

translating, in response to the identified cut vertex set and the security policy statements, into configuration data for security functions operating at nodes in the cut vertex set; and

establishing the configuration data in the security functions at the nodes in the cut vertex set.

First Hit

□ Generate Collection

L15: Entry 19 of 19 File: DWPI Jun 22, 1995

DERWENT-ACC-NO: 1995-231777
DERWENT-WEEK: 199746

TITLE: Object-orientated rule-based protocol processing for distributed computer networks - selects appropriate configuration for type and quality of service using distributed network directory and naming services

Basic Abstract Text (3):
USE/ADVANTAGE - For operation of distributed, heterogenous, wide area computer network, allowing dynamic protocol configuration, with rules defined for each protocol family for translating type/quality of service into particular common protocol set for synchronous data stream transactions.

PF Application Date (1):
19940411

PF Application Date (2):
19940411

PF Application Date (3):
19940411

PF Application Date (4):
19940411

PF Application Date (7):
19940411

PF Application Date (9):
19940411

PF Application Date (10):
19940411

PF Application Date (12):
19931217

PF Application Date (13):
19940411

PF Application Date (14):
19940411

PF Application Date (16):
19940411

PF Application Date (17):
19940411

First Hit Fwd Refs

□ Generate Collection

L11: Entry 1 of 3 File: USPT Nov 19, 2002

DOCUMENT-IDENTIFIER: US 6484261 B1
**** See image for Certificate of Correction ****
TITLE: Graphical network security policy management

Application Filing Date (1):
19981211

Brief Summary Text (34):
One feature of this aspect is that the step of defining the set of symbols includes the steps of displaying the set of symbols in a window of a user interface; receiving user input from a user input device coupled to the user interface, in which the user input defines how to manipulate the symbols to create the symbolic representation of the policy. Another feature is that the step of defining the information communication policy includes the steps of receiving editing commands for re-configuring the symbolic representation; re-configuring the one or more symbols into a revised symbolic representation of the policy based on the editing commands; and displaying the revised symbolic representation on the user interface. In a related feature, the step of re-configuring the one or more symbols includes the steps of automatically validating the editing commands according to one or more syntactic rules and based on the context of the editing commands.

CLAIMS:

14. The method as recited in claim 1, wherein the step of defining the information communication policy includes the steps of: receiving editing commands for re-configuring the symbolic representation; re-configuring the one or more symbols into a revised symbolic representation of the policy based on the editing commands; and displaying the revised symbolic representation on the user interface.

First Hit Fwd Refs

□ Generate Collection

L11: Entry 2 of 3 File: USPT Jul 27, 1999

DOCUMENT-IDENTIFIER: US 5930759 A
TITLE: Method and system for processing health care electronic data transactions

Application Filing Date (1):
19960430

Detailed Description Text (40):
If in response to the prompt of FIG. 9A, the operator indicates that the patient has secondary insurance, processor 22 then shows the screen of FIG. 8B on the screen. This screen contains a column of words or phrases, each of which represents a standard question or command. For instance, the word "name" represents the command: "Enter the name of the secondary insurer," and the phrase "Group #" represents the command: "Enter the group number of the policy of the secondary insurer."

First Hit Fwd Refs

□ Generate Collection

L14: Entry 2 of 4 File: USPT Oct 9, 2001

DOCUMENT-IDENTIFIER: US 6301613 B1
TITLE: Verifying that a network management policy used by a computer system can be satisfied and is feasible for use

Application Filing Date (1):
19981203

Brief Summary Text (7):
Each device 102 stores information about its current configuration, and other information, in one or more forms, for example, a Management Information Base (MIB) 114. Information in the MIB 114 is organized in one or more MIB variables. The network management station 10 can send "fetch" and "set" commands to the device 102 in order to retrieve or set values of MIB variables. Examples of MIB variables include sysObjectID and sysDescr. For information stored in other forms, there are other types of communications and commands to set and retrieve the information values.

Detailed Description Text (15):
The Configuration Understanding element 202 is a means to read the relevant parts of the configuration of equipment and services that are to be managed by the policy system and to translate idiosyncratic representations into standard forms that the rest of the system can deal with. In this context, "configuration" is the state of a network device, which may be discovered in a variety of ways. The "standard forms" are formally defined, machine-processable, representations of the idiosyncratic representations contained in the actual equipment. The form, content and meaning of the standard form representations are all formally defined. Generally there is a many to one relationship between idiosyncratic and standard forms. Only the representation standardization mechanism of the system must deal with the idiosyncrasies and, then, only to translate to a standard form. All other components of the system read and understand the standard forms. The "standard forms" may be one or more rules or tables that define the configuration of devices in the network. For example, the configuration of a network may be stored as values in tables of a relational database system, in which each table is associated with a characteristic of a device. The configuration information may also be stored in persistent object structures in computer memory.

Detailed Description Text (23):
In block 222, the process carries out a configuration understanding step. Block 222 may involve invoking one or more external processes that discover information about devices in a managed network. For example, a discovery process can use SNMP commands to query devices in the network and receive copies of values stored in their MIBs, for example, one or more MIB variable values. Block 222 also involves converting the values that are received into a standard format that can be accessed and understood by other steps in the process of FIG. 2B. This may involve, for example, storing the values in a configuration file 230, or storing the values in an object model in memory. It is desirable, but not required, to have a complete, standard representation of configuration information that is stored in a way that can be uniformly retrieved by other steps of the process of FIG. 2B. Block 222 may also involve carrying out the functions described above with respect to Configuration Understanding element 202.

Detailed Description Text (31):

Computer system 300 may be coupled via bus 302 to a display 312, such as a cathode ray tube (CRT), for displaying information to a computer user. An input device 314, including alphanumeric and other keys, is coupled to bus 302 for communicating information and command selections to processor 304. Another type of user input device is cursor control 316, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 304 and for controlling cursor movement on display 312. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

First Hit Fwd Refs

□ Generate Collection

L14: Entry 3 of 4 File: USPT Dec 28, 1999

DOCUMENT-IDENTIFIER: US 6009081 A
**** See image for Certificate of Correction ****
TITLE: Private network access point router for interconnecting among internet route providers

Application Filing Date (1):
19970903

Detailed Description Text (107):
A second step 189 in the flow chart of FIG. 11 is responsible for creating the necessary "base" router configurations for each P-NAP Provider. These base configurations (base configurations plus additional configuration added by the processes of the present invention) are located on network management server 144 of FIG. 1 as well as in the respective routers 105, 106 and 116. A base configuration for a particular P-NAP Provider will only be located on the router to which that P-NAP Provider connects. The primary purpose of this step is to translate the routing policy guidelines of the first step 188 into appropriate router configuration commands. Where appropriate, flags will be added to the base router configurations to tie the base configurations into the added configuration commands of further steps.

Detailed Description Text (119):
A third step 190 in the flow chart of FIG. 11 is responsible for creating the "deny" list of AS numbers for each provider, turning that list into the appropriate router configuration commands adding the commands to the base router configuration and filling in the base configurations with the AS list pointer value (noted as <filled in by third step 190> above).

Detailed Description Text (121):
The resulting "deny" configuration commands for UUNet might be:

Detailed Description Text (258):
An eighth step 195 in the flow chart of FIG. 11 is that of causing the router to apply the P-NAP Provider configuration files to the routes. With Cisco routers, in the preferred embodiment, this is done through the router configuration commands which were created in the preceding steps 188, 189, 190, 191 and 192.

Detailed Description Text (261):
The next step in the series of steps in FIG. 12 is indicated by a box 202 labeled "For each Provider". Box 202 indicates that the path verification process will occur for each of the Providers shown in FIG. 2 and listed above. Within the remainder of this process, the term "Provider" is used to indicate the current Provider. The next step in the series of steps is indicated by a box 203 which is labeled "prtraceroute to Provider traceroute server". The box 203 thus indicates that the command "prtraceroute" is invoked from the server Network Mgmt-1 144 (FIG. 3) to the current Provider traceroute server. It will be understood that "prtraceroute" is an enhanced version of the "traceroute" application which appends AS numbers to each hop. In the preferred embodiment, such a command used to trace to the current Sprint traceroute server would return the following output:

Detailed Description Text (326):
The sequence then flows to a box 288 labeled "Add all AS.sub.-- PATH "deny" statements to Provider specific configuration file". This step takes all of the previously discussed "deny" statements and adds them to the Provider specific configuration. The sequence then flows to a decision block 289 labeled "Additional other Providers?". If there are additional Providers to process which are not the outer loop Provider, then the answer to this decision block would be yes, and the sequence would flow back to box 285. If the answer to decision block 289 is no, then the sequence flows to a box 290 labeled "Search Exception As database for all Exception AS values". Prior to box 290, the inner loop beginning at box 285 and ending at decision block 289 implemented the previously discussed concept of taking the union of the Provider AS Data database 181 and subtracting the AS numbers of the current Provider. Once at box 290, the Exception AS Data database 182 is queried for all of its data. The sequence then flows to a box 291 labeled "Create an AS.sub.-- PATH "deny" statement for each AS value using unique list number". This creates "deny" statements in the same manner as box 287, using the same unique list number (because we are still within the same outer loop Provider). The sequence then flows to a box 292 labeled "Add all AS.sub.-- PATH "deny" statements to Provider specific configuration file". This step is the same as box 288. The sequence then flows to a box 293 labeled "Put unique list number into the "match as-path" of appropriate route-map". As discussed previously, each Provider configuration has a Cisco configuration called a "route-map" which is labeled <Provider>-LOCAL-PREF. Within this route map is a "match as-path <fill in the blank>" command. The step of box 293 places the unique access list number of box 284 into this "match as-path . . . " command. As an example, the MCI LOCAL.sub.-- PREF route-map would look like the following:

Detailed Description Text (328):
FIG. 18 is a flow chart indicating the steps involved in adding the appropriate number of AS additions to routes advertised to each Provider. FIG. 18 is the detail of the AS.sub.-- PATH length configuration of step number four 191 of FIG. 11. This process starts at a terminal 300 labeled "Begin". The first step is indicated by a box 301 labeled "For each Provider". This is the beginning of a loop which will process once for each P-NAP Provider. There is no requirement that the Providers be processed in a specific order. The sequence then moves to a box 302 labeled "Retrieve number of AS additions for this Provider". Thus, a search of the Provider AS.sub.-- PATH Prepend Data database 186 is made for the current Provider to find the number of AS additions which need to be added for this Provider. As was described previously, the Provider AS.sub.-- PATH Prepend Data database 186 is populated by the algorithm described in connection with FIG. 15. The sequence then moves to a box 303 labeled "Add this number of P-NAP ASes to Provider configuration". Within each Provider configuration, as was discussed previously, there exists a Cisco route-map named "<Provider>-ASPATH-PREPEND". Within this route-map there is a "set as-path prepend <fill in the blank>" command. At the <fill in the blank> point should be a number of the current P-NAP AS numbers (6993 in the P-NAP of the present invention) equal to the number retrieved from the Provider AS.sub.-- PATH Prepend Data database 186. Thus, using the P-NAP of the present invention, The ANS-ASPATH-PREPEND route-map would look like the following: